

TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse Instituut

Infotehnoloogia eriala

Timo Kallaste

Õppematerjalide loomine rakendustarkvarale Project Anarchy

Bakalaureusetöö (6 EAP)

Juhendaja: Anne Villems

Kaasjuhendaja: Taavi Duvín

TARTU, 2014

Lühikokkuvõte

Õppematerjalide loomine rakendustarkvarale Project Anarchy

Käesoleva bakalaureusetöö eesmärgiks on luua õppematerjalid rakendustarkvarale Project Anarchy. Viimane on tarkvara, mis on mõeldud eelkõige mobiilsete mängude loomiseks. Töö koosneb kolmest peatükist, millest esimeses selgitatakse modernse mängumootori ülesehitust ja kirjeldatakse kahte mängumootorit. Teises peatükis analüüsitakse e-kursust, selle ülesehitust ning keskkonda, milles kursus toimub. Kolmandas peatükis antakse detailsem ülevaade loodava kursuse struktuurist.

Märksõnad: Project Anarchy, mängud, mängude loomine, õppematerjalid, mängumootor.

Abstract

Creating tutorial series for Project Anarchy

The aim of this bachelor thesis is to create tutorial series for Project Anarchy SDK. Tutorials are targeted for mobile devices because this is the aim of the Project Anarchy SDK. The tutorial series are combined into one online course in which a game is developed using Project Anarchy. The material consists of three parts. The first part gives an overview of the modern game engine architecture and describes two game engines in detail. The second part analyses online course. Describes the target group and goals. It also concentrates on the online learning platform that is used for the course. The third part gives an overview of the tutorial set's architecture. Every topic in tutorial set is covered in detail.

Keywords: Project Anarchy, games, game development, game engine, tutorial.

Sisukord

Sissejuhatus.....	4
1. Mobiilsetele seadmetele mõeldud mängumootorite ülevaade.....	6
1.1 Mängumootor.....	7
1.2 Mängumootori ehitus.....	8
1.2.1 Juhtprogrammi arhitektuur.....	8
1.2.2 Tööriistad.....	11
1.3 Näited Mängumootoritest.....	12
1.3.1 Project Anarchy.....	12
1.3.2 Unity.....	13
1.3.3 Project Anarchy ja Unity õppimine.....	14
2. Kursuse ülevaade.....	15
2.1 Kursuse eesmärk.....	15
2.1.1 Kursuse analüüs.....	15
2.1.2 Olemasolevad kursused Project Anarchy kohta.....	16
2.2 Õpiväljundid.....	16
2.3 Kursuse vorm.....	16
2.4 Sihtrühma analüüs.....	17
2.5 Hindamine.....	18
2.6 Juhendmaterjal.....	18
2.7 Läbiviimine.....	19
3. Mängu loomine Project Anarchy vahenditega.....	21
3.1 Kursuse ehitus.....	22
3.1.1. Sissejuhatus.....	22
3.1.2. Uue projekti loomine ja Anarchy ülevaade.....	22
3.1.3 Maailma loomine.....	22
3.1.4 Kontrollitava karakteri loomine.....	24
3.1.5 Mängu loogika ja füüsikasimulatsiooni tegemine.....	25
3.1.6 Mängu testimine.....	25
Kokkuvõte.....	26
Viited.....	27
Lisad.....	30
Lisa 1. Kursuse materjalid.....	30

Sissejuhatus

Videomängutööstuses on seoses viimasel ajal kasvanud mobiilsete seadmete levikuga tekkinud uus haru. Firmed, kes toodavad ja avaldavad suure käibega mängu personaalarvutitele ning konsoolidele, on hakanud looma mängu ka mobiilsetele seadmetele. Ubisoft, kes on tuntuks saanud Assassin's Creed mängude seeriaga, tuli välja Apple mobiilsetel seadmetel mänguga Assassin's Creed Pirates. Enne seda tõid nad turule Assassin's Creed IV, mis oli suunatud mängukonsoolidele ja personaalarvutitele [8]. See näitab, et mobiilidele suunatud videomängud on tuntud mängude tootjatele üheks lisaturuks.

Aastaks 2013 jõudis avalikkuse ette mitu edukat mängu, mida olid tootnud iseseisvad (*indie*) arendajad. Iseseisvaks arendajaks loetakse inimest või gruppi inimesi, kes ei saa märkimisväärset finantsilist tuge väljastpoolt. See tähendab, et rakendus arendatakse enda vahenditega. Ühena võib nimetada mängu Fez, arendajaks Polytron Corporation, mida oli müüdud 2013 aasta lõpuks miljon koopiat [1]. Edulood üksikuritajatest mängude loomisel on inspiratsiooniks teistele arendajatele, näidates sellega, et ei pea olema suure finantseeringu ja arendusmeeskonnaga, loomaks videomängu, mis jõuaks sadade tuhandete mängijateni.

Nagu iga asja ehitamiseks, on ka videomängude loomisel vaja tööriistu. Olgu nendeks siis graafiliste liidestega WYSIWYG (*what you see is what you get*) redaktorid või mingi programmeerimiskeele teekide kogum. Mängu tüüp määrab ära, millist tööriista kasutada. Videomängude tüüpe on palju erinevaid. On rallimängud, kus mängija eesmärk on läbida autoga etteantud rada. Siin võib tüüpilise näitena välja tuua firma EA poolt avaldatud seeria Need for Speed [9]. On platvormmängud (*platformers*), kus mängija peamine eesmärk on jõuda ettemääratud punkti liikudes ühelt platvormilt teisele. Näiteks Nintendo poolt loodud Super Mario Bros [10]. Ära tuleks märkida veel esimeses isikus tulistasmäng (*First Person Shooter*), milles mängija näeb maailma läbi peategelase silmade. Selle žanri üheks näiteks on Battlefield 4 [11]. Videomängude stiile on veel nagu näiteks rollimängud, kus mängija arendab karakterit erinevate oskustega, et olla tugevam. Oluliseks aspektiks tööriistade valikul on sihtplatvormi valik. Mobiilsete seadmete puhul on enim kasutatavateks operatsioonisüsteemideks Google poolt loodud Android ning Apple seadmetel iOS [12]. Mõlemad operatsioonisüsteemid töötavad nii mobiiltelefonidel kui tahvelarvutitel. Mängumootori valimisel on tähtis arendaja varasem kogemus. Kogemus ühel platvormil annab eelise selle näol, et teatakse süsteemi ehitust ja nõudeid. Neljandaks kasutatava tööriista litsents. On olemas avatud lähtekoodiga mängude loomise raamistikud nagu näiteks Cocos-2D [13]. Alternatiivina on visuaalse kasutajaliidesega mängumootorid nagu Unity, kus tuleb maksta litsentsitasu siis, kui soovitakse lisaosasid kasutada. Näiteks selleks, et asendada mängu algsel laadimisel Unity logo enda omaga, tuleb soetada pro litsents [14]. Viiendaks on oluline dokumentatsioon ja kasutajabaas antud vahendil. Suurem hulk kasutajaid annab parema võimaluse saada vastuseid tekkivatele küsimustele. Dokumentatsioonile lisaks võivad olla eraldi õppematerjalid. Need aitavad kiiremini ja süstematiseeritumalt omandada teatud oskust. Näiteks Unityl on selleks otstarbeks eraldi

õppematerjalide (*tutorials*) sektsioon. Seal on võimalik valida mängustiil, mida soovitakse luua ning tuuakse välja tehnikad, mida ja kuidas kasutada [15].

Antud töö on loodud eesmärgiga teha õppematerjal platvormmängu loomiseks. Vahendiks on valitud tööriistade kogumik nimega Project Anarchy [16]. See on 2013. aastal turule toodud tööriistade kogum eelkõige mobiilsete mängude tegemiseks. Havok, firma Project Anarchy taga, on ennast tõestanud vahevara loomisega mängudele. Vahevaraks (*middleware*) loetakse tarkvara, mis ühendab muidu kahte või enam eraldiseisvat rakendusprogrammi. Seda võib vaadelda kui silda, mida kasutatakse, et liigutada andmeid kahe programmi vahel, mis pole muidu omavahel ühilduvad. Havok on firma, mis on teinud veel Havok Physics füüsika simulaatori, Havok AI tehisintellekti vahevara ning programmi Havok Cloth, mis on riidest materjali simulatsioonimootor.

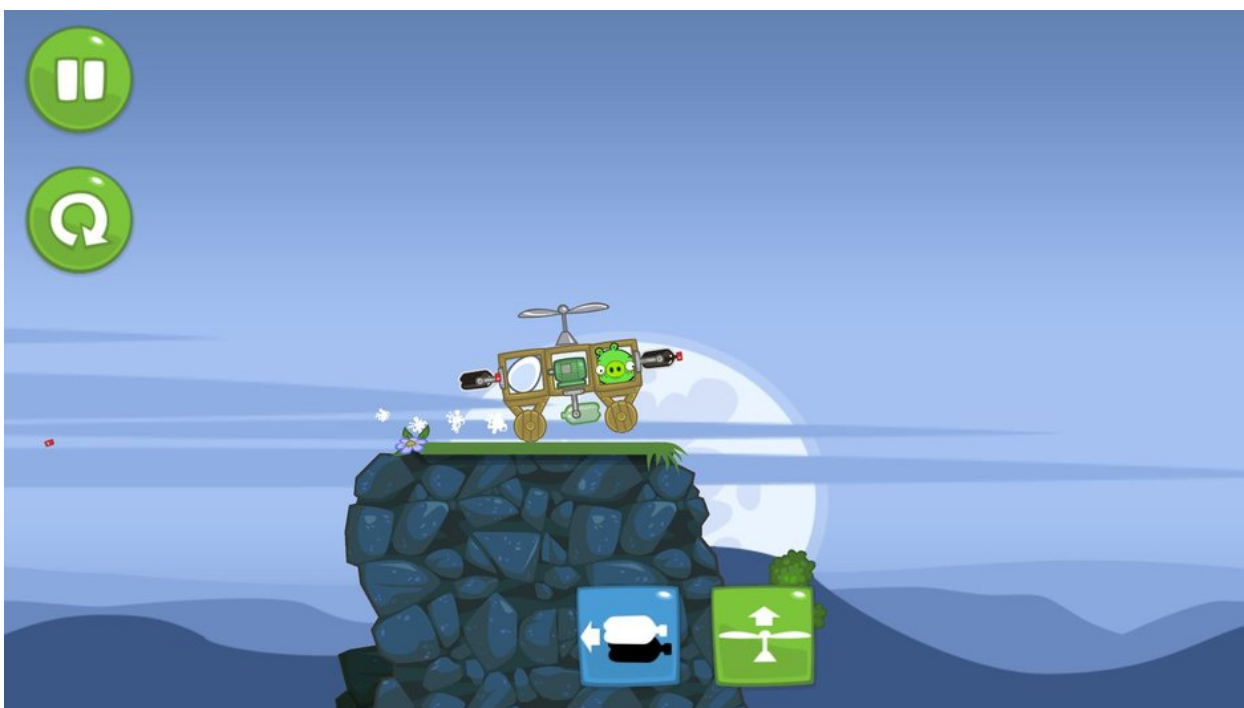
Käesolev töö on jaotatud kolmeks peatükiks. Esimeses peatükis antakse ülevaade mängumootoritest. Kirjeldatakse nende ehitust ning tööpõhimõtteid. Teoreetilise ülevaate paremaks mõistmiseks on välja toodud detailsemad ülevaated Project Anarchy ja Unity kohta.

Teises peatükis on ära toodud ülevaade veebikursuse loomisest. Selles analüüsitakse kursust, toetudes e-kursuse loomise juhendile [40].

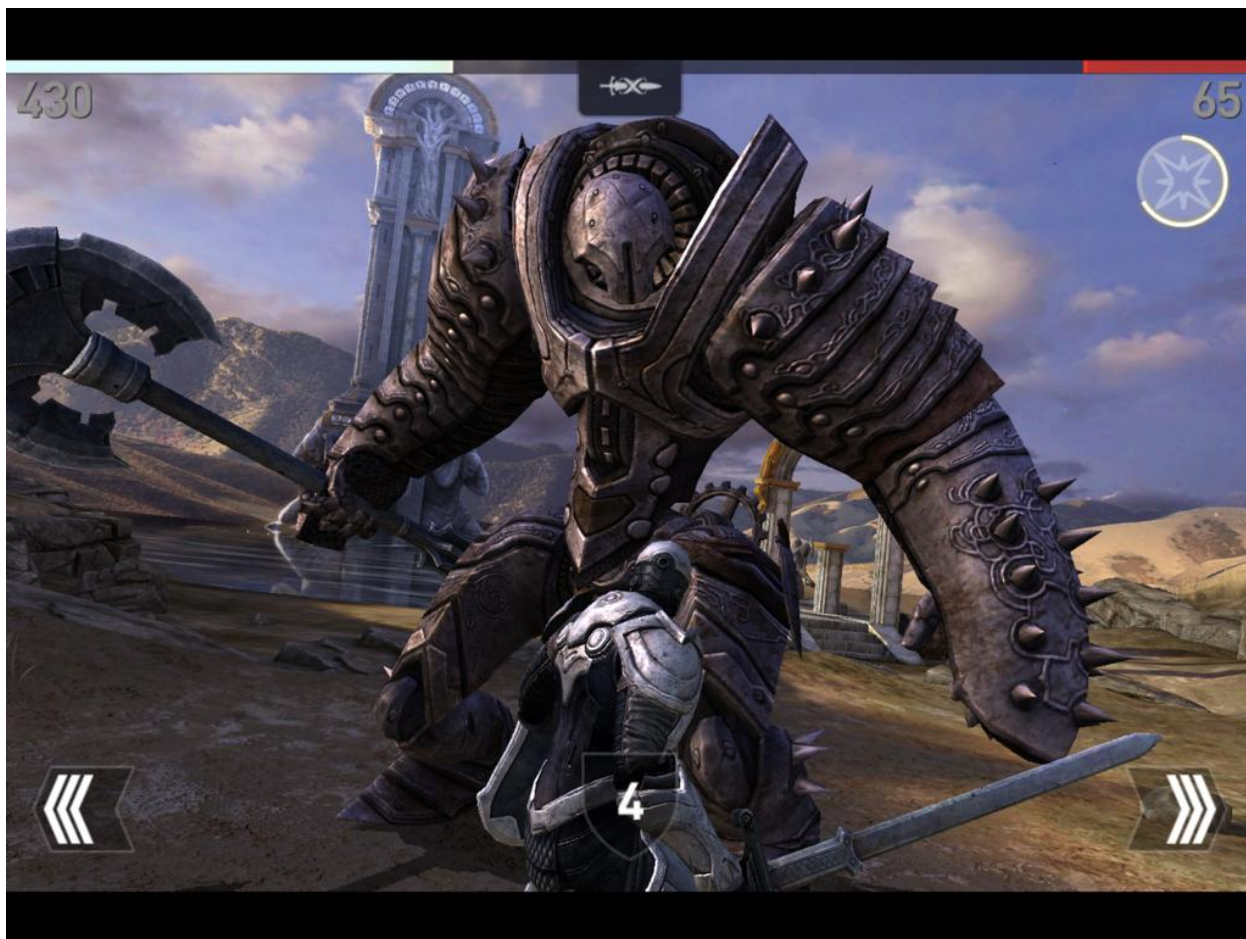
Kolmandas peatükis on ülevaade antud töö raames loodavast kursusest Project Anarchy kohta. Kirjeldatakse kursuse struktuuri ning õppematerjale. Tuuakse välja eesmärgid ja tähtsamad ülesanded iga loengu kohta.

1. Mobiilsetele seadmetele mõeldud mängumootorite ülevaade

Mobiilsete mängude osakaal digitaalsetes poodides nagu Google Playstore ja Apple Appstore, on tõusnud märgatavalt [17]. Mängude varieeruvus on suur. Näiteks Bad Piggies, mis on kahemõõtmelise graafikaga füüsika simulatsioonil põhinev mäng. See on arendatud Soome firma Rovio poolt ja kasutab Unity mängumootorit (joonis 1.0) [3]. Teistsuguse näitena võib tuua graafiliselt (joonis 1.1) silmapaistva ja kriitikute poolt hästi vastu võetud mänguga Infinity Blade 3 [39]. See on Chair Entertainmenti ja Epic Games-i koostöö vili. Epic Games on mängumootori Unreal Engine arendaja ning antud mäng on tehtud demonstreerimaks selle mootori võimekust mobiilsetel seadmetel [4].



Joonis 1.0 - Bad Piggies [18]



Joonis 1.1 - Infinity Blade III [19]

Joonistel 1.1 ja 1.0 on näha väga erineva põhimõttega mängu. Lisaks on ka nende visuaalne pool täiesti erinev üksteisest. Nende mängude tegemisel on kasutatud erinevaid mängumootoreid. Mängumootorite iseärasuste mõistmiseks tuleb kõigepealt teada, mida see endast kujutab.

1.1 Mängumootor

Mängumootor on süsteem mängude loomiseks ja arendamiseks [2]. Seda võib vaadata kui auto mootorit. Juhul pole vaja teada, kuidas tema auto mootor täpselt töötab. Ta teab, et kui keerata võtit, siis auto peab käivituma ja sellega saab liikuda. Mängumootori põhimõte on sama. Eesmärk on ühendada väiksemaid mootoreid, mis vastutavad igaüks enda valdkonna eest ja siduda need kokku üheks komplektiks. Näiteks helide haldamise mootor. Helifaili lisamiseks mängu on vaja see lugeda arvuti kõvakettalt või mälust, pakkida vajadusel kokku ja lahti ning mängida õigel hetkel. See ongi helimootori ülesanne. Mängumootor koosneb paljudest kindla eesmärgiga osadest ja koondab need ühte raamistikku või rakendusse kokku.

Oletame, et eelmises lõigus mainitud autojuht otsustas enda punase auto siniseks värvida. Selleks läks ta töökotta ja värvis auto kere siniseks. Antud protsess ei mõjuta mitte kuidagi mootori tööd, sest ta pole teadlik, mis värvi raam teda ümbritseb. Sama kehtib modernses mängumootoris. See ei tohi olla seotud otseselt mängu enda loogikaga [20]. Mängumootori ülesanne on olla piisavalt abstraktne, et võimaldada ehitada selle sama süsteemiga erinevaid mängu. Näiteks Project Anarchy puhul on kasutusel Havok Rigid Body objekt. Selle objekti ülesanne on reageerida füüsikasimulatsioonis toimuvatele muudatustele. Kui programmis simuleeritakse tuult, siis Havok Rigid Body võimaldab defineerida objekti, mida tuul mõjutab. Selleks võib olla puuleht või plastiktõps, mis lendab tuulega. Järelikult on mängumootoris olemas abstraktne komponent, mida saab kasutada, et teha mänguspetsiifilisi objekte.

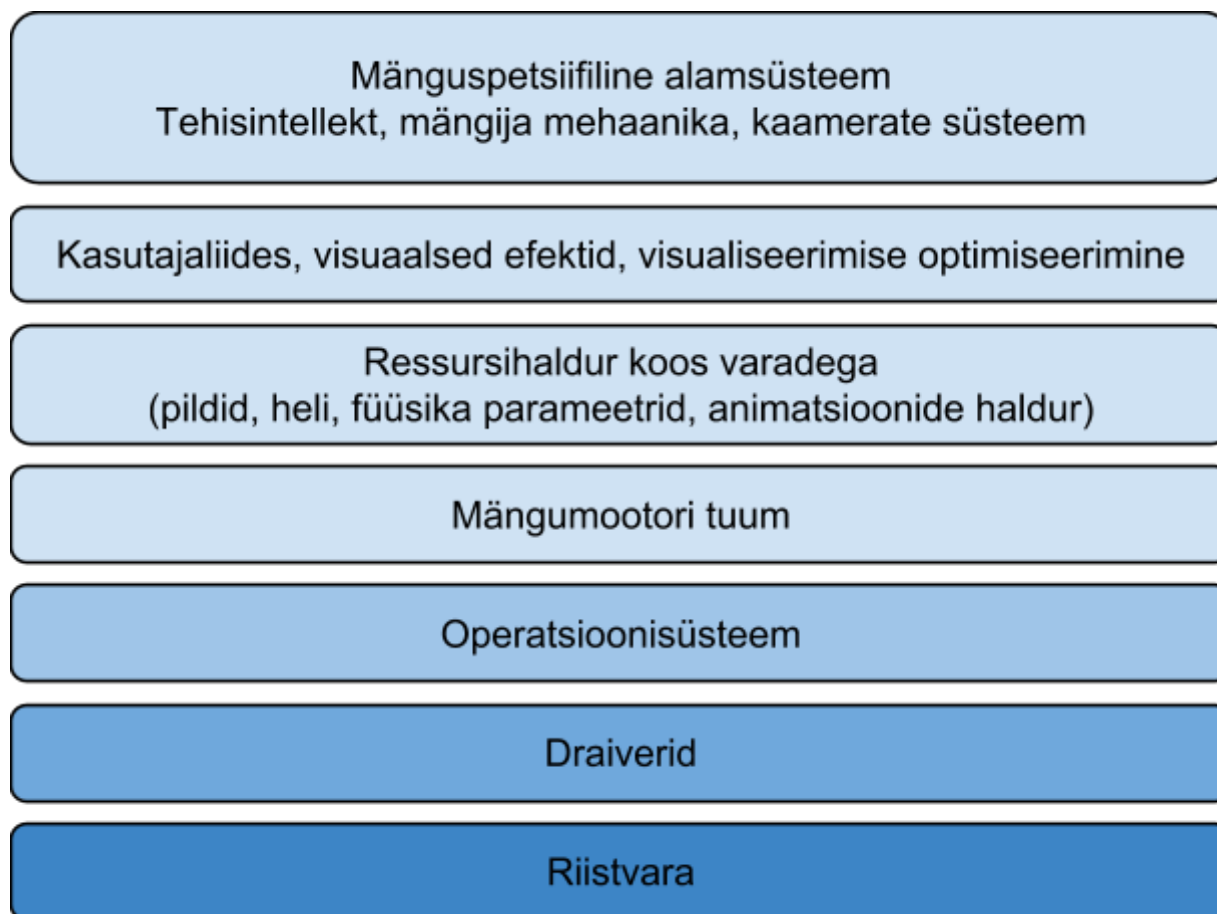
1.2 Mängumootori ehitus

Mängumootor sisaldab endas kahte osa - tööriistad, mis hõlbustavad mängude ehitamist ja juhtprogramm ehk *runtime* [20]. Tööriistade osast räägitakse peatükis 1.2.2 Tööriistad. Esmalt vaadeldakse juhtprogrammi osa. Kui mäng on valmis tehtud ja see käivitatakse, alustabki tööd juhtprogramm. See käib kuni mängu sulgemiseni. Tema ülesandeks on hallata erinevaid spetsiifilisi mootoreid samal ajal, kui mäng käib. Näiteks mängumootor teab, et tal on kontroll helimootori üle ning kui tuleb käsk, et on vaja mängida heli, kasutab ta antud mootorit selleks otstarbeks. Kokkuvõtlikult on tegu vahekihiga, mis teab, millised spetsiifilisemad mootorid on tema käsutuses ja millal neid kasutada.

1.2.1 Juhtprogrammi arhitektuur

Mängumootorid ehitatakse kihtidena [20] (lk). Tüüpiliselt ülemised kihid sõltuvad alumistest, aga mitte vastupidi. Mootorid, mis on suunatud strateegiamängudele, on disainitud teistmoodi, kui rallimängudele mõeldud vahendid. Rallimängudes on vaja tekitada tunnetus kiirest liikumisest ja auto peab käituma vastavalt füüsikaseadustele. Strateegiamängudes on vaja kuvada palju objekte korraga ekraanil, et mängija saaks neid kontrollida. Siiski on mõlemale žanrile disainitud mängumootorites vaja kasutada mingisugust füüsikaseaduste simulatsiooni. Strateegia- ja rallimängudes on vaja teha objektide kokkupõrke kontrolli. Selle käigus võrreldakse mängu objekti teiste mänguobjektidega ning kontrollitakse, kas esimene võib teisest läbi liikuda või on tegu läbitamatu objektiga. Vastavalt tulemusele tehakse muudatused mängu sees. Kui auto sõidab vastu seina, arvutab füüsikasimulaator välja, kui suured peaksid olema auto purustused. Ilma kontrollita saaks ralliautoga sõita teiste mängijate autodest läbi ja strateegiamängus inimesed liiguksid ehitistest läbi.

Mängumootorite spetsialiseerumise mingile spetsiifilisele stiilile määrabki pisemate komponentide olemasolu. Selleks kasutatakse nende lisamist ja eemaldamist kihtidest. Mudelil 1.3 on ära toodud kõik peamised juhtprogrammi komponendid modernses mängumootoris. Joonis on lihtsuse mõttes üldine, kuid rakendub peaaegu kõikidele mängu tüüpidele. Järgnevalt räägitakse lähemalt igast komponendist.



Mudel 1.3 - Juhtprogrammi komponendid

a) Riistvara kiht - Riistvara kiht määrab ära ressursid, mida on võimalik kasutada. Tänapäeval on täiesti tavaline, et mobiilsel seadmel on 1-2 gigabaiti mälu ja 2-4 tuumaline protsessor [7]. Lisaks erinevad liidesed nagu puutetundlik ekraan, güroskoop ja liikumisandur [21]. Kaasaegne mängumootor on teadlik, milliseid liideseid saab kasutada antud seadmel ning võimaldab ligipääsu nendele läbi draiveri ja operatsioonisüsteemi kihi.

b) Draiveri kiht - Seadme draiverid on madala taseme tarkvarakomponendid. Need võivad olla tehtud konkreetse riistvaratootja või operatsioonisüsteemi enda poolt. Draiverite eesmärk on hallata ja eraldada ülemiste kihtide ning riistvara vaheline suhtlus [20] (lk).

c) Operatsioonisüsteemi kiht - Operatsioonisüsteemi ülesandeks on juhtida mitme programmi tööd ühes seadmes. Üheks selliseks programmiks on ka mäng. Mäng peab arvestama operatsioonisüsteemi poolt loodud reeglitega. Nende alla kuuluvad näiteks mäluhaldusega seotud piirangud ja lõimede haldamine. Androidi ja iOS peal jooksevad reeglina mitmed rakendused korraga ja operatsioonisüsteem on see, mis kontrollib, et kõik saaks enda toimingud tehtud. Kui

mõni rakendus ei suuda jääda ettemääratud reeglistikku, näiteks võtab endale liiga palju mälust, siis suletakse ta operatsioonisüsteemi poolt [22].

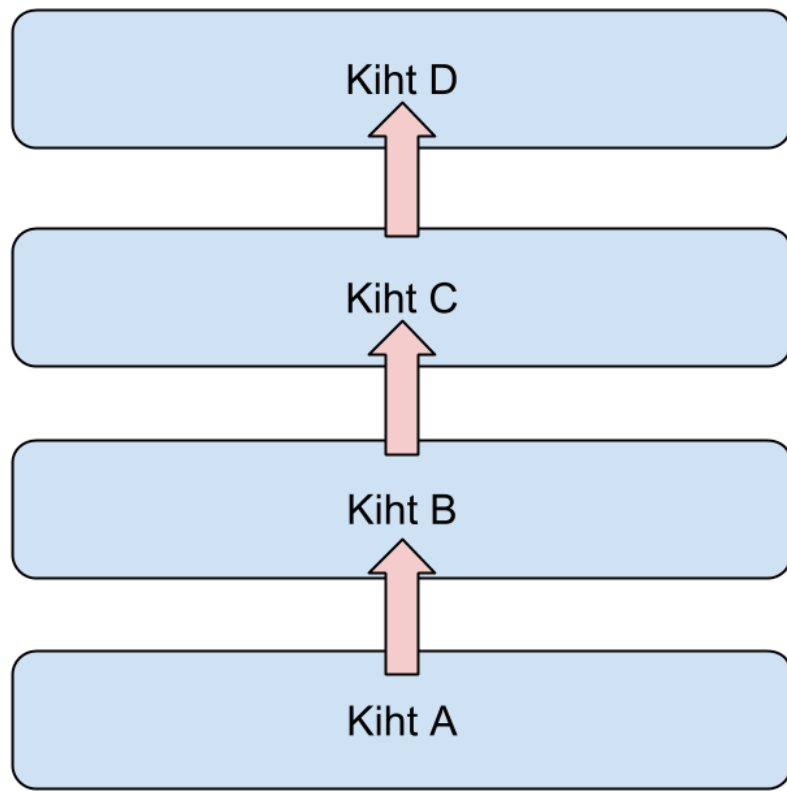
d) Mängumootori tuum - Selles kihis tehakse valmis üldine süsteem, mille eesmärk on jooksutada mängu. Lähtestatakse (*initialise*) sõnumite vahetamise moodul, mille ülesandeks on mängu erinevate objektide vahel infot vahetada. Näiteks kui mängija hüppab ühelt platvormilt teisele ja maandub sellel platvormil, siis saadetakse animatsioonihooldurile teade, et mängija on nüüd maa peal ja seega on vaja muuta tema hüppamise animatsioon hoopis kõndimise animatsiooniks.

Tuum sisaldab endas kõiki mänguobjekte. Esiteks staatilised objektid nagu taust või ehitised. Teiseks on seal ära kirjeldatud dünaamilised objektid nagu mängija poolt kontrollitav objekt. Dünaamiline tähendab antud kontekstis seda, et tema mingeid parameetreid on võimalik muuta. Näiteks mängija vajutab klaviatuuril nuppu mille tagajärjel mängu tegelane hakkab enda positsiooni muutma ehk liikuma.

Tähtsaim komponent selles blokis on skriptimise võimalus. Mängumootori tuum on sild madala taseme komponentide ja kõrge taseme skriptimiskeele vahel. Skriptidega on võimalik teha spetsiifiline mänguloogika enda rakendusele.

e) Ressursihaldur koos varadega - Iga keerulisem tarkvararakendus vajab hulga utiliite. Utiliitide eesmärk on lihtsustada mingisugust toimingut [23]. Tüüpiliselt on need seotud ressursihaldusega. Näiteks kettalt andmete lugemine. Lisaks võib ära mainida veel mäluhalduse ja matemaatilised teegid. Samuti on ressursihalduri ülesanne stseenide kuvamise optimeerimine. Üheks optimeerimise võimaluseks on objektide sorteerimine mingite parameetrite põhjal. Tüüpilisemaks on kaugus mängijast. Kaugel olevaid elemente ei arvestata ning need eemaldatakse kuvamise arvutustest.

Kihiline arhitektuur võimaldab tehnika arenedes mingeid kihte uuendada. Kuna ainult ülemine kiht sõltub alumisest siis on võimalik alumine kiht selliselt ära vahetada, et ülemine saaks enda jaoks vajaliku info kätte. Oletame, et joonisel 1.4 on vaja muuta kihti C. Tuleb rõhutada, et ülemised kihid sõltuvad alumistest, aga mitte vastupidi. Selleks, et kihti C saaks ehitada, peab ta ühilduma kihiga B. Seega ei pea ta teadma tegelikult kihist D mitte midagi. Kiht D ehitatakse tema otsa. See tähendab seda, et ühe osa muutmiseks on see vaja ühildada ainult ühe osaga. Programmis, milles on palju kihte, on selline arhitektuur väga kasulik ja võimaldab modifitseerida olemasolevat arhitektuuri.



Joonis 1.4 - Kihiline arhitektuur

1.2.2 Tööriistad

Eelmises peatükis näidati, millised ülesanded on juhtprogrammil. Nende hulgas oli ka ressursihaldur. Meeldetuletusena võib öelda, et selle peamine eesmärk on hallata mängusiseseid vahendeid nagu graafika ja heli. Tööriistade alla kuulub ka skriptimiskeele redaktor, kui see on mängumotooris olemas. Redaktori ülesandeks on lihtsustada skriptimist. Üheks kasulikumaks lihtsustamise viisiks on automaatne sõne lõpetamine. Kui kasutaja tahab kirjutada meetodi nime `onGameStart`, siis peale “onG” kirjutamist pakutakse talle nimekirja erinevatest meetoditest, mis algavad sellise kolmetähelise järjestusega.

Graafika ja heli puhul on oluline, et neid oleks võimalik kiirelt ning ilma funktsionaalsete kadudeta mängumootorisse importida. Tihti tehakse näiteks 3D graafika selleks otstarbeks mõeldud tarkvaras. Tasuta tarkvaradest saab siinkohal ära märkida programmi Blender, mida saab kasutada nii animatsioonide, kui staatilise graafika tegemiseks. Olles valmistanud graafikaobjekti, lihtsuse mõttes olgu selleks praegusel juhul kast, tuleb see Blenderist eksportida

ning importida mängumootorisse. Selleks puhuks ongi loodud tööriistad, et eksportida kast kõikide detailidega Blenderist välja ning importida mängumootorisse.

Käesolevas töös on tööriistade osa kaetud põgusalt. Üheks põhjuseks on nende suur hulk. Igal mängumootoril on spetsiaalselt selle programmi jaoks tehtud tööriistade komplekt, mida kasutatakse. Järgnevas peatükis on kirjeldatud mõned olemasolevad mängumootorid. Muu hulgas kirjeldatakse seal olevaid tööriistu.

1.3 Näited Mängumootoritest

Käesolevas punktis antakse ülevaade kahest mängumootorist. Üheks on Project Anarchy ja teiseks Unity.

1.3.1 Project Anarchy

2013 aasta suvel tuli mängude jaoks tarkvara tootev firma Havok turule mobiilsete mängude loomiseks mõeldud vahendite kogumikuga nimega Project Anarchy [24]. See sisaldab endas Havok Vision Engine nimelist mängumootorit ning erinevaid tööriistu. Tasuta litsentsiga on lubatud arendada mängu järgmistele platvormidele: Android, iOS ja Tizen [25]. Tizen on Samsungi poolt loodud mobiilne operatsioonisüsteem [26].

Havok on pannud enda tehtud tarkvara ühte gruppi kokku. Kogu seda vahendite hulka nimetatakse ühise nimega Project Anarchy. Sinna alla kuuluvad Havok Vision Engine, mis on mängumootor, millel Project Anarchy põhineb. Havok Physics on kokkupõrgete arvutamiseks ja füüsikaliste simulatsioonide jaoks loodud tarkvara. Animatsiooni haldamise tööriistaks on Havok Animation Studio. Tihti on mängudes tarvis leida lühim teekond (*pathfinding*) punktist A punkti B. Selle probleemi lahendamiseks on Havok välja töötanud tööriista nimega Havok AI, mis pakub lahendusi erinevate teekondade leidmiseks [5].

Project Anarchy tööriistu on võimalik laiendada enda kirjutatud moodulitega, kasutades programmipaketti Project Anarchy SDK (*Software Development Kit*). See on C++ keeles kirjutatud arendustarkvara, et võimaldada mängumootori laiendamist arendajate enda vahenditega.

Vision Engine olemasolevaid võimalusi saab täiendada kasutades skriptimist. Mängumootoriga on kaasas redaktor Lua keeles skriptimiseks [27]. Lua on programmeerimiskeel, mis ongi loodud eelkõige skriptimiseks. Lua on kirjutatud madala taseme programmeerimiskeeles C [38].

Project Anarchy allalaadimine on tasuta. Seda võib kasutada kommertslikul eesmärgil ilma litsentsitasuta. Sel juhul, peab arendaja panema programmi sisse nähtavale kohale Havok'i logo. Litsentsitasud lisanduvad siis, kui soovitakse rakendus välja lasta muule platvormile, kui Android, iOS või Tizen [5].

Havok tõi turule mängu Anarchy RPG, mis on tehtud Project Anarchy vahenditega [28, 29]. Tegu on kolmanda isiku vaates rollimänguga. Mängija kontrollib ühte karakterit, keda ta näeb ekraanil ning saab liigutada või anda käskke vastaste löömiseks. Rollimäng on loodud selleks, et näidata Project Anarchy võimalusi. Arendajad saavad proovida seeläbi valmis toodet aitamaks otsustada, kas seda kasutada enda projektis või mitte. Kolmandate osapoolte tehtud mängu praegu veel avalikustatud pole.

1.3.2 Unity

Unity on mängumootor, mida on arendanud samanimeline firma. Esimest versiooni tutvustati aastal 2005 [6]. Praeguseks on see jõudnud versioonini 4.3.4. Selle mängumootoriga on võimalik arendada järgmistele mobiilsetele operatsioonisüsteemidele: Android, iOS, Windows Phone ning Blackberry OS [30]. Windows Phone on Microsofti poolt tehtud mobiilne operatsioonisüsteem. Blackberry on Blackberry Ltd poolt loodud operatsioonisüsteem, mis jookseb selle firma mobiilsetel seadmetel. Unity üks eeliseid võrreldes Project Anarchyga, millega saab arendada vaid Windowsi operatsioonisüsteemi kasutades, on see, et seda saab kasutada ka Apple OSX operatsioonisüsteemil [31].

Unity on visuaalse redigeerijaga mängumootor, mis tähendab, et kasutaja näeb kohe tulemust, kui on tehtud muudatus mootoris. Näiteks muudetud mõne mängu objekti suurust või värvi. Unity hõlmab endas animatsioonitehnoloogiat nimega Mecanim. Selle tööriista ülesandeks on hallata animatsioone. Näiteks üleminekut karakteri kõndimise animatsioonist jooksmiseks [32].

Unity on tasuta allalaetav. Tasuta litsentsi korral tuleb kasutada Unity enda laadimisekraani, kus on nende logo. Iga platvormi jaoks on vaja soetada selle spetsiifiline litsents, mis avab tasuta versioonis lukus olevad võimalused [33]. Näiteks iOS korral avab pro litsents videote mängimise võimaluse programmis sees ning samuti striimimise võimaluse.

Skriptimiskeelteks on C# ja Unityscript. Viimane on Javascripti põhjal ehitatud skriptimiskeel, et täiustada ja lisada olemasolevat funktsionaalsust [34].

Seda mängumootorit kasutades on tehtud mängu strateegiamängudest nagu Blackguards kuni kosmosesimulatsioonini Kerbal Space Program [35]. Blackguards on Daedalic Entertainment GmbH poolt loodud käigupõhine strateegiamäng. Eesmärgiks on vastase üksused vallutada, juhtides enda üksuseid [36]. Kerbal Space Program on mäng firma Squad poolt, kus eesmärgiks on ehitada kosmoserakett, millega lennata kuule ja tagasi. Selleks otstarbeks on kasutusel realistlik füüsikasimulatsioonimootor, millega on püütud luua võimalikult tõetruu päris maailma simulatsioon.

1.3.3 Project Anarchy ja Unity õppimine

Project Anarchy on saadaval olnud alles 11 kuud [24]. Selle aja jooksul pole ilmunud palju materjale antud tarkvara õppimiseks. Eelmises punktis on ära toodud Unity ja Project Anarchy ülevaade. Nende põhjal võib väita, et Project Anarchy sisaldab endas kõiki tänapäevaseid võimalusi mängude loomiseks.

Peamiseks erinevuseks Project Anarchy ja Unity vahel on kasutajate arv. Unity'l on 2 miljonit registreerinud arendajat. Selline arendajate hulk on kogunenud 9 aastaga. Havok pole avalikustanud andmeid kasutajate hulga kohta, kuid arvestades turul oldud aega, võib arvata, et arendajate arv on oluliselt väiksem. Üheks põhjuseks võib olla õppematerjalide kättesaadavus, mis on Unity puhul väga hästi organiseeritud firma enda poolt. Lisaks kasutajate õpetused ning kolmandate osapoolte poolt loodud kursused mängumootori selgeks õppimiseks. Arvestades Project Anarchy potentsiaali ning odavat litsentsi võrreldes Unityga, otsustas selle bakalaureusetöö autor teha õppematerjalid just Project Anarchy kohta.

Selles peatükis tutvustati lugejale mängumootorite üldist ülesehitust. Lisaks toodi ära kaks konkreetset vahendit mängude loomiseks - Project Anarchy ja Unity. Järgmisest peatükist alates hakatakse tegelema probleemiga kuidas koostada efektiivset kursust mängude loomiseks. Seda Project Anarchy näite abil.

2. Kursuse ülevaade

Esimeses peatükis anti ülevaade mängumootorite ehitusest. Lisaks toodi näidetena välja Project Anarchy ja Unity. Käesolevas peatükis kirjeldatakse e-kursuse loomise põhimõtteid. Räägitakse e-õppe vahendist, mida kasutatakse kursuse tegemisel. Analüüsitakse sihtgruppi ning kirjeldatakse kursuse eesmärki.

2.1 Kursuse eesmärk

Eelmises peatükis räägiti mängumootoritest üldiselt. Võib väita, et tegu on keeruliste programmidega. Palju erinevaid mooduleid probleemide lahendamiseks lisab keerukust programmile ning selle õppimisele. Käesoleva peatüki aluseks on “Juhend kvaliteetse e-kursuse loomiseks” [40]. See juhend on loodud ühtlustamaks arusaama kvaliteetsest e-kursusest. Antud bakalaureuse töö kasutab juhendit, et luua kvaliteetne e-kursus Project Anarchy vahendite õpetamiseks.

2.1.1 Kursuse analüüs

Iga uue rakendustarkvara kasutamisel tuleb esmalt õppida seda tundma. Vahendeid programmis asuvate tööriistade selgitamiseks on mitmeid. Inimene võib kasutada katse ja eksituse meetodit. Selles proovitakse erinevad funktsionaalsused läbi ning jälgitakse resultaati. Vastavalt parameetrite muutmisele võib muutuda ka tulemus. Resultaadi põhjal on võimalik otsustada, kas seda funktsionaalsust on tarvis kasutada või mitte.

Teiseks võimaluseks on lugeda programmiga kaasasolevat dokumentatsiooni. Dokumentatsiooni ülesanne on selgitada tarkvaras leiduvat funktsionaalsust. Project Anarchy installeerimisel antakse kasutajale valida, kas soovitakse dokumentatsiooni alla laadida või mitte. Kogu materjal on jagatud gruppideks vastavalt otstarbele. Eraldi on ära toodud dokumentatsioon programmeerijatele ning kunstnikele. Muu hulgas leidub veel *Vision Engine* sissejuhatav manuaal ja õpetused. Manuaalid on laiahaardelisemad ning käsitlevad tarkvara kõiki või enimkasutatud funktsioone.

Õpetuste eesmärk on näidata lahenduskäiku eelnevalt püstitatud probleemile. Näiteks on olemas õpetus, kuidas lisada mängu animeeritud tegelasi. Selles näidatakse sammude kaupa, mida on vaja teha, et sisestada enda mängu animeeritud tegelane. Võrreldes manuaaliga, võib pidada efektiivsemaks õppimise vormiks õpetusi, sest siis antakse ülevaade ainult nendest tööriistadest, mida kasutada antud ülesande lahendamisel. Konkreetse ülesandega kitsendatakse materjali hulka. See tähendab, et õpilane ei pea lugema tervet manuaali läbi teadmaks, kuidas ülesanne lahendada. Näiteks õpetus, kuidas luua Project Anarchy vahenditega mäng. Õpilane saab vajalikud baasteadmised, kuidas mingit konkreetset mängu luua. Peale materjali omandamist on õpilasel olemas piisav ülevaade kogu programmist, et luua enda mäng, mis ei pea olema õpetuses olnuga sarnane.

2.1.2 Olemasolevad kursused Project Anarchy kohta

Hetkel on olemas videoõpetuste seeria “Zero-to-hero”, mille väljaandjaks on Havok ise. Selles videoseerias tehakse lihtsa loogikaga mäng algusest lõpuni. Eesmärgiks on palliga puudutada ekraani üleval servas olevaid plokkke. Puudutamisel need kaovad väljakult. Kui kõik blokid on kadunud, on mängija võitnud. Pall allub füüsilise maailma simulatsioonile, mis paneb ta pörkama. Gravitatsioon tõmbab palli ekraani alumise serva suunas. Mängija kontrollib risttahukat, mida ta saab liigutada paremale või vasakule, et takistada palli ekraani alla kukkumast. Kui pall pörkab vastu kontrollitavad risttahukat, pörkab ta ülespoole. Kui pall kukub ekraani alumisse serva, kaotab mängija ühe palli. Kokku on 5 palli. Kaotades kõik pallid, on mäng läbi.

Peale “Zero-To-Hero” õpetuste seeria on Havok teinud videoõpetusi konkreetsete probleemide lahendamiseks. Need on jaotatud järgmistesse kategooriatesse: programmeerimine, kunst/animatsioon ja mängumaailma disain. Kategooriates olevad õpetused pole otseselt omavahel seotud ja ei loo terviklikku õpikeskkonda erinevalt “Zero-to-hero” õpetuste seeriast.

Praegu puudub platvormmängu õpetus Project Anarchy vahenditele. Platvormmängudes kasutatakse tehnikaid, mida saab teistesse mängu stiilidesse üle kanda. Näiteks kui karakter soovib tulistada relvast kuuli, on vaja luua kuuli objekt. Siis määratakse kuuli objektile positsioon ning suund. Platvormmängudes on vaja kasutada mingisugust objektide kokkupõrke kontrolli. Mängu tegelane ei tohi kukkuda läbi maapinna, kui tegemist pole erijuhuga. Platvormmängus omandatud teadmisi saab hiljem teistesse mängu stiilidesse edasi kanda. Nii tulistamist, kui objektide kokkupõrke kontrolli saab kasutada tulistasmängudes. Seepärast on platvormmängu programmi õppimiseks kasulik.

2.2 Õpiväljundid

Loodava kursuse eesmärgiks on anda õpilasele ülevaade mängude loomisest kasutades Project Anarchy vahendeid. Kursuse raames valmib algusest lõpuni platvormmäng. Õpilane oskab luua struktureeritud projekti kasutades Project Anarchy vahendeid. Ta omab baastadmisi mängude loomisest ning saab rakendada neid enda projektidel. Tähtsaimaks õpiväljundiks on baastadmiste loomine, mille põhjal on õpilasel võimalik ehitada enda platvormmäng.

2.3 Kursuse vorm

Kursus on loodud Udemy veebiõppe keskkonda. Udemy on veebikeskkond, mis sisaldab endas üle 12000 kursuse. Sellel on 2 miljonit registreeritud õpilast. Kursuseid leidub mitmetest valdkondadest. Alates joogast ning lõpetades tuumafüüsikaga [37]. Udemy keskkonda saavad kasutada kõik, kellel on ligipääs internetile.

Peamiseks põhjuseks, miks valiti Udeemy keskkond on võimalus jõuda paljude kasutajateni vähesel reklaamiga. Selles veebiõppe keskkonnas on võimalik otsida kursuseid kategooriate ja märksõnade kaupa. Teiseks põhjuseks on lihtne kursuse loomine. Tuleb luua kursus. Seejärel saab nende alla läbi veebiliidese lisada õppematerjalid ja küsimustikud.

Kursuste hinnad sõltuvad kursusest. Selle määrab õppematerjalide autor. Kursuste loomine on tasuta. Keskkonna kasutamise eest tuleb tasuda vaid juhul, kui loodav kursus on tasuline. Siis tasutakse õppija poolt makstud summast 50% Udeemyle.

Udeemy võimaldab jagada kursuse sektsioonideks ja need omakorda loenguteks. Loengute all võivad olla videoloengud või -õpetused ning nendega seonduvad materjalid. Oluline on, et kõik oleks loogilise järjestusega ja konsistentne. Viimase all mõeldakse seda, et raskusaste ei varieeruks, ülesehitus oleks neil sarnane ning kaetav õppematerjali maht oleks sama. Sellega välditakse olukorda, kus ühe peatüki omandamine võtab teisest oluliselt kauem aega.

Lähtudes Udeemy ülesehitusest on käesoleva töö raames tehtav kursus iseseisvaks õppimiseks. Selle raames pole määratud konkreetset instruktorit, kes juhendaks. Probleemide korral on võimalik läbi Udeemy pöörduda kursuse looja poole.

Project Anarchy kohta loodav kursus on struktuurilisest loogikast lähtuv. See tähendab, et õppematerjalide järjestus on dikteeritud aine loogilisest struktuurist. Videomängude loomist saab vaadelda kui ühte järjestikust protsessi, mis on jagatud etappideks. Kusjuures uus etapp on sõltuv vana olemasolust. Selleks, et lisada mängu graafikat, on vaja, et seda oleks võimalik mängu sees kuvada ekraanil. Loomulikult saab teha asju paralleelselt. Kuna antud kursus on eelkõige töövahendi tutvustamiseks ühe tervikliku näite põhjal, siis lihtsuse mõttes on loengute järjestus tehtud sedasi, et mäng ehitatakse järk-järgult valmis. Näiteks graafika lisatakse mängu alles siis, kui selle jaoks on olemas loogika ja seda on võimalik kasutada.

Kursusel kasutatakse videomaterjali, kus räägitakse asjadest üldisemalt ja keskendutakse pigem mängu ehitamisele kasutades Project Anarchy vahendeid. Lisaks on iga teema kohta ka kirjalikke materjale. Viimastes on välja toodud olulisemad põhimõtted käsitletava teema kohta.

2.4 Sihtrühma analüüs

Antud kursuse sihtrühmaks on õpilased, kes tahavad kasutada Project Anarchy vahendeid, et luua oma mäng. Nende eesmärgiks peaks olema valmis teha mäng otsast lõpuni. Kursus on inglise keeles - seega on nõutav inglise keelest arusaamine.

Õpilastelt eeldatakse head arvuti kasutamise oskust. Kursusel kasutatakse operatsioonisüsteemi Microsoft Windows 8.1. Õpilane võib kasutada operatsioonisüsteemi Windows 7, mis on Windows 8 eelkäija. Kasutaja peab olema võimeline alla laadima ja iseseisvalt installeerima Project Anarchy tööriistad.

Kursusel kasutatakse programmeerimist. Seepärast oodatakse õpilaselt mõningat programmeerimiskogemust. Selle puudumisel võib skriptimise osa muutuda liiga keeruliseks. Väheste programmeerimise kogemusega õpilasel on vaja rohkem vaeva näha, et baasoskused endale selgeks teha. Selleks on lisatud viited abistavatele materjalidele. Kogenumale õpilasele võivad mõned teemad olla liiga kerged. Alati on võimalus liikuda järgmise loengu juurde. Siiski on soovitatav enne sooritada iga teema lõpus olev test, et saada teada, kas antud materjali põhilised punktid on omandatud.

2.5 Hindamine

Kursusel puudub määratud instruktor, kes kontrolliks ülesandeid. Seepärast on tähtis, et õpilasel oleks alati võimalus enne järgmise peatüki juurde liikumist ennast ise kontrollida. Iga peatüki lõpus on ülesanded ja järgmise peatüki alguses saab alla laadida projekti uuest seisust. Õppija ülesandeks on ära teha puuduvad osad vastavalt ülesannetele. Kui õpilane jääb hätta, on olemas suunavad materjalid. Kui ka need ei aita, on võimalus alla laadida projekti selline seis, kus ülesanded on lahendatud. See annab õpilasele võimaluse alati edasi liikuda järgmise peatükini.

2.6 Juhendmaterjal

Kuigi peamisteks materjalideks on videloengud, on olemas ka kirjalikud materjalid. Ainult videoõpetustega ei pruugita tervet materjali piisavalt hästi edastada. Kirjalike materjalidega on võimalik lisada skeeme ning selgitada neid. Sel juhul on kasutajal võimalik õppida enda tasemele vastava kiirusega, ilma, et ta peaks videot seisma panema või tagasi kerima. Sellega on võimaldatud õpilasel valida endale sobivam tempo.

Kirjalikest materjalidest on kergem infot otsida. Videomaterjali on vaja kerida õigesse kohta ning selle leidmine võib olla aeganõudev. Tekstifailist on otsimisfunktsiooniga võimalik leida vajalik info vaid märksõna kirjutamisega. Üheks sihtgrupiks on inimesed, kes loovad selle kursuse raames enda esimese mängu. Nendele on väga oluline, et oleks võimalik hõlpsalt materjalide seast infot leida. Videomaterjalides kõikide terminite lahtiseletamine teeb neid pikemaks. Kogenumate õpilaste jaoks võivad need kohad muutuda igavaks, sest korratakse juba tuttavaid tehnikaid ja termineid.

Kirjalike materjalide üheks plussiks on see, et seda on võimalik igal ajal lugeda. Videoloengu jaoks on tarvis kuulata heli ning see seab otsese piirangu kasutamiseks. Õpilasel peab olema võimalik kuulata loengut, et seda mõista. Kirjaliku materjali puhul on oluline vaid see, et ekraanil saaks infot kuvada.

Videomaterjalide ülesandeks on detailselt näidata, milliseid vahendeid kasutada. Samuti rääkida põgusalt, miks nii toimitakse. Eesmärgiks on püstitada konkreetse peatüki kohta ülesanne, lahendada see loengu jooksul ära ning näidata tulemust. Selle läbi antakse õpilasele ülevaade, milliseid tööriistu kasutatakse.

2.7 Läbiviimine

Kursusel osaleja saab ise vabalt valida aja, millal ta soovib materjali omandada. See võimaldab õpilasel õppida enda tempoga. Iga teema õppimise järel on võimalus kohe asuda järgmise juurde.

2.8 Õpilasteni jõudmine

Õppematerjalidest on kasu vaid sel juhul, kui nad on kättesaadavad ja hästi leitavad. Eriti oluliseks on see muutunud tänapäeval, kui internet on täis erineva tasemega kursuseid. Sealt on vaja välja paista, et jõuda õpilasteni. Selleks puhuks on Udemy enda sees kasutusel soovitamise mehhanism ning otsingumootor. Soovitamise ülesandeks on leida kasutajad, kes võivad olla antud teemast huvitatud. Näiteks kui õpilane on varem võtnud mängude loomisega seotud kursuseid, pakutakse talle mängudega seotud kursuseid ka teistelt autoritelt.

Soovitamine ei pruugi alati töötada. Seda juhul, kui kasutaja teab konkreetsemalt, mida ta soovib õppida. Selleks juhiks on Udemy keskkonnas olemas otsingumootor. Sisestades märksõnad kuvatakse kursuste nimekiri, milles leiduvad otsitavad sõnad.

Eelnevad kaks võimalust nõuavad kasutajalt juba mingisugust sisendit. Nad peavad olema millestki varem juba osa võtnud või vähemalt huvi tundnud. Teisel juhul peab olema õpilane teadlik, mida ta otsib. Udemy pakub lisaks eelnevaile võimaluse enda kursusega silma paista kõigile, kes seda veebikeskkonda külastavad. Kriteeriumiteks on kiire kursusel osalejate arvu kasv ning kõrge keskmine hinne. Selle läbi on võimalus jõuda inimesteni, kes ei ole varem huvitatud mängude tegemisest, kuid märkavad kursust Udemy veebisaidil [42].

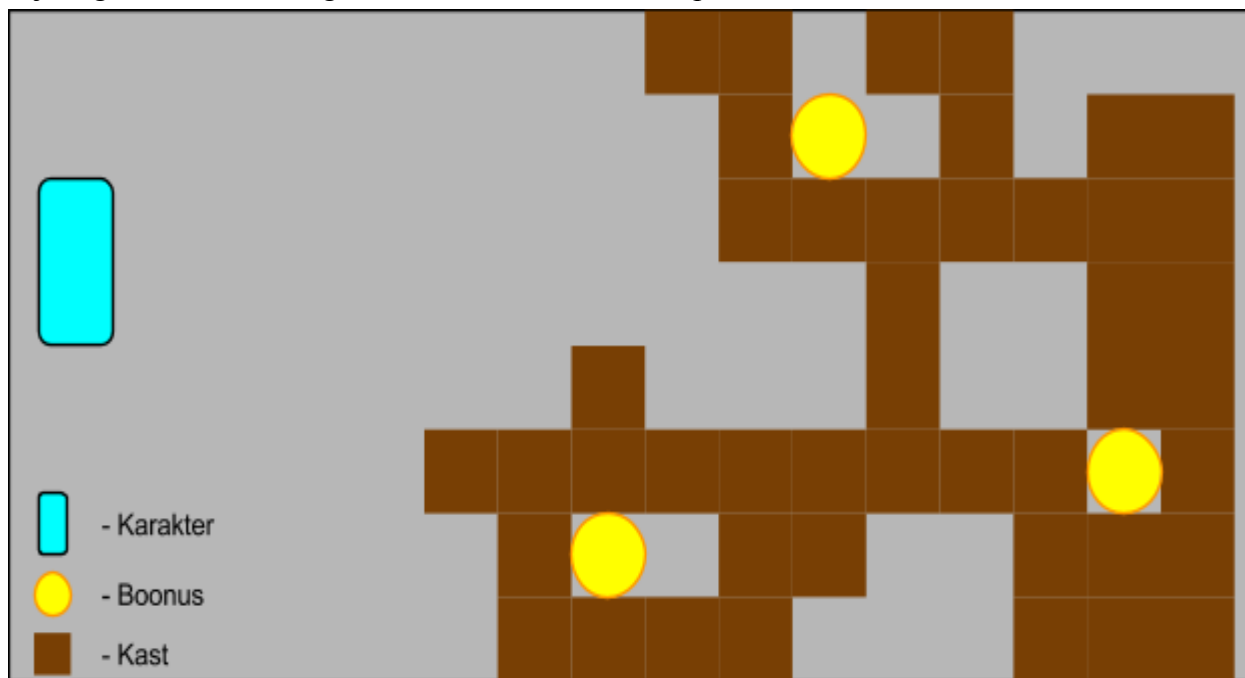
Huvi mängude loomise vastu on kasvanud viimasel ajal. Seda seoses mängude tegemise vahendite odavnemisega ning õppematerjalide kättesaadavuse suurenemisega. Project Anarchy jaoks pole loodud palju terviklikke kursuseid, kus näidataks, kuidas luua mäng otsast lõpuni. On olemas Havok'i poolt loodud "Zero-To-Hero" seeria. Selline olukord, kus inimesed on huvitatud, kuid materjalide hulk on võrreldes konkurentidega väiksem, on loonud hea võimaluse avaldada kursus Project Anarchy kohta. Väikene konkurentide hulk suurendab võimalusi enda materjalidega silma paista.

Käesolevas punktis toodi välja üldised põhimõtted kvaliteetse e-kursuse loomisel. Antud töö käigus loodav kursus tehakse Udemy veebiõppe keskkonnas. Peamiseks põhjuseks on paljude õpilasteni jõudmine ning kursuse ülesseadmise lihtsus. Järgmises peatükis on kirjeldatud loodavat kursust lähemalt. Lugejale antakse ülevaade teemadest ning üldisest kursuse struktuurist.

3. Mängu loomine Project Anarchy vahenditega

Eelnevates peatükkides kirjeldati mängumootorite ehitust ja põhimõtet ning kursuse loomist Udemy keskkonnas. Käesolevas peatükis on kirjeldatud täpsemalt kursust, milles luuakse videomäng kasutades Project Anarchy vahendeid. Peatükk on jaotatud punktideks. Igale lõigule vastab kursusel olev peatükk. Igas peatükis käsitletakse uut teemat.

Kursuse raames tehtava mängu idee on koguda punkte. On liikuv objekt, karakter, mida kontrollib mängija (Joonis 3.1). See objekt saab liikuda ekraanil kas üles või alla. Lisaks on liikumatud objektid. Nendeks on kastid ja boonused. Kastid ja boonuse objektid tekivad juhuse alusel paremal pool ekraani ning liiguvad vasakule poole ekraani. Karakterit on võimalik juhtida üles ja alla, et vältida kokkupõrget kastidega. Kui toimub kokkupõrge kastiga, vähendatakse mängija elupunkte. Mängijal on alguses 5 elupunkti. Kokkupõrke vältimiseks on lisaks liikumisele võimalik karakteriga tulistada kastide pihta kuule, mille peale need purunevad. Punkte saab koguda tulistades kuuli boonuse objekti pihta või kui karakter puutub boonuse objektiga kokku. Eesmärgiks on saada maksimaalselt punkte.



Joonis 3.1

Mäng on mõeldud puutetundliku ekraaniga seadmele. Project Anarchy on suunatud eelkõige mobiilsetele seadmetele. Viimaste seas valitsevad turgu just puutetundliku ekraaniga nutiseadmed, olgu nendeks mobiiltelefonid või tahvelarvutid. Kursuse jooksul tuuakse välja puutetundliku ekraani eripärad võrreldes traditsiooniliste sisendseadmetega nagu arvutihiir või klaviatuur.

Kursus on jaotatud teemadeks. Kokku on kuus teemat. Kursus algab sissejuhatusega ning lõpeb valmis mängu testimisega mobiilsel seadmel.

3.1 Kursuse ehitus

3.1.1. Sissejuhatus

Kursuse raames luuakse Project Anarchy vahenditega mäng. Kursus algab projekti ülevaatega. Selles räägitakse üldiselt, milline peaks olema projekti struktuur. Pannaks paika eesmärgid, et kasutaja teaks, mida oodata. Näidatakse lõpptulemust ja terve kursuse struktuuri. Sissejuhatav peatükk on mõeldud selleks, et kasutajat motiveerida.

Sissejuhatuse eesmärgiks on kursust reklaamida. Tegu on ainukese peatükiga, mis on avalik ja kättesaadav kõigile huvilistele. On tähtis, et selle videoloengu lõpus omaks õpilane selget nägemust kursusest. Selle põhjal on tal võimalik teha valik, kas ta soovib osaleda või mitte. Reklaam peab olema aus. Kasutajale näidatakse vaid neid materjale, mida käsitletakse kursuse teistes loengutes.

3.1.2. Uue projekti loomine ja Anarchy ülevaade

Esimeses sisulises peatükis räägitakse Project Anarchy vahenditest üldiselt. Kasutaja suunatakse Project Anarchy kodulehele, et sa saaks alla laadida vastava vahendi. Peale seda antakse põgus ülevaade vForge kasutajaliidesest. Tegemist on Vision Engine mängumootori redigeerimise vahendiga. Õpilasele näidatakse, kus asuvad põhilised funktsioonid. Põhiliste funktsioonide all mõeldakse selle kursuse jooksul enimkasutatud funktsioone.

Ülevaate andmise järel alustatakse uue projekti loomisega. Siin kasutatakse Project Anarchy juurde kuuluvat funktsionaalsust nimega “Game solution creator”. Viimane võimaldab luua uue projekti struktuuri. Struktuuri all on silmas peetud projekti jaotamist gruppideks. Näiteks luuakse grupp, milles hoitakse skripte ja teine grupp, milles hoitakse graafikaga seotud objekte. Grupeerimine on oluline projekti haldamiseks. Kui projekt sisaldab palju objekte, on gruppidest kergem otsida ühte vajaminevat.

Projekti loomise järel tehakse loengus kokkuvõtte läbitud materjali kohta. Õpilasele meenutatakse, et tal on võimalik teha teadmiste kontroll kasutades loengu materjalides olevat küsimuste faili. Lisaks meenutatakse kirjalike materjalide olemasolu. Materjalides on ära toodud läbitud loengu kohta käivad selgitused ning viited lisainfo hankimiseks. Lõpetuseks antakse ülevaade järgmisest peatükist.

3.1.3 Maailma loomine

Projekti loomise järel saab alustada maailma ehitamist. Teine sisuline peatükk on mõeldud selleks, et ehitada valmis mängu maailm. Maailm hõlmab endas kõiki mängu objekte ning nendevahelist loogikat. See tähendab, et on üks objekt, mis haldab teiste objektide vahelist seost.

Objektidevahelise seoste kirjeldamiseks kasutatakse relatsioonimaatriksit. Tabelis 3.1 on ära toodud selline maatriks. Seal on ära kirjeldatud kõik objektid, mis luuakse kursuse raames. Märgend “x” tähendab, et kahe objekti vahel on seos. Näiteks objekt “kuul” ja “kast A” omavad seost, sest nad sisaldavad märgendit. Antud juhul on selleks seoseks kasti purunemine ning kuuli eemaldamine mängumaailmast. Kui seos puudub kahe objekti vahel, on lahter tühi. See tähendab, et sellist olukorda ei hallata. Näiteks tegelase ja kuuli vahel pole vaja seos, sest nad ei puutu kokku.

	Kast A	Kast B	Tegelane	Boonus	Kuul
Kast A			x		x
Kast B			x		x
Tegelane	x	x		x	
Boonus			x		x
Kuul	x	x		x	

Tabel 3.1 - Objektidevahelised seosed

Maatriksi selgitusele järgneb õpetus objektide importimise kohta. Selles teema osas näidatakse, kuidas olemasolevaid mängu elemente tuua projekti sisse ja neid selles kasutada. Antakse ülevaade, milliseid parameetreid muuta.

Nüüdseks on projektis olemas kõik vajalikud graafikaobjektid. Järgmise sammuna pannakse paika vaade, mida mängus kasutatakse. Selleks luuakse kaamera objekt ja kasutatakse “Set to camera position” funktsiooni. Nimetatud funktsiooni ülesandeks on salvestada hetke kaamera positsioon redigeerimisvaates. Kui kasutaja alustab mängimist, seatakse mängumaailma kaamera salvestatud positsioonile.

Võttes aluseks disainidokumendi, mis antakse antud teema tekstimaterjalidega kaasa, alustatakse mängumaailma ehitamist. Disainidokument kirjeldab ära mängu kujunduse ning loogika. Mängumaailma ehitamiseks paigutatakse kastide ja boonuste objektid vastavalt disainidokumendis määratud positsioonidele.

Viimase ülesandena luuakse valgustus mängumaailmale. Selle teema käsitluses räägitakse üldisemalt mängumaailmate valgustamise põhimõtetest. Õpilasele selgitatakse sellist tehnikat nagu valguse salvestamine tekstuuridesse (*lightmapping*). Valguse salvestamine tekstuuridesse tähendab seda, et objekti olemasolevale tekstuurile lisatakse valguse info. Oletame, et mängumaailmas on objekt, mis on kujult kuubik ja punast värvi tekstuuriga. Lisame samasse maailma valgusallika, mis on suunatud täpselt kuubiku pealmisele tahule. See tahk jääb ainukesena valgustatuks. Selle kuubiku punast tekstuuri muudetakse selliselt, et ainult peamise tahu tekstuur jääb punaseks, sest see on valgustatud. Teised jäävad mustaks, sest nendeni valgus ei jõudnud.

Valgustamise osaga lõpetatakse teine peatükk. Loenguga on kaasas kirjalik dokument, milles seletatakse mõningaid peatükis käsitletud teemasid põhjalikumalt. Materjalide hulgas on kontrolltest kasutajale, mida ta saab iseseisvalt sooritada.

Teise loengute seeriaga on õpilasele näidatud, kuidas luua mängumaailm Project Anarchy vahenditega. Talle on selgitatud, kuidas importida graafika objekte, sättida paika kaamera ja paigutada staatilisi mänguobjekte redigeerimisvahendiga.

3.1.4 Kontrollitava karakteri loomine

Loodav platvormmäng koosneb peale staatiliste objektide ka dünaamilistest objektidest. Dünaamilisi objekte võib vaadelda kui objekte, mis muudavad enda olekut. Oleku muutuse põhjustab kokkupuude mõne teise objektiga või mingi sisend. Sisendiks võib olla kasutaja poolt antud käsklus liikuda mingis suunas.

Loodavas mängus on karakteril 3 liikumisseisundit ja üks eriline seisund. Liikumisstaadiumiteks on liikumine üles ja alla ning seismine antud positsioonil. Eriline staadium on kuulide tulistamine mingis suunas.

Kasutaja saab anda sisendeid, et muuta karakteri seisundeid. Vajutades sõrmega ekraani vasakul poolel alustab karakter liikumist samale y-koordinadile, kus on sõrm. Vajutades ekraani paremal poolel sõrmega alustab karakter kuulide tulistamist. Kuuli liikumistee arvutatakse karakteri positsiooni ja ekraani paremal pool oleva sõrme positsiooni põhjal.

Projekti arendamisel on hõlpsam testida mängu selles arvutis, mida kasutatakse arendamiseks. Iga muudatuse järel testimine mobiilsel seadmel lisab alguses keerukust. Põhjuseks on projekti konverteerimine mobiilse seadme jaoks sobivasse vormi. Sellega tegeletakse viimases loengus, kus õpetatakse mängu jooksutamist iOS ja Android seadmetel.

Testimine arendatavas arvutis tähendab seda, et lisaks puutetundliku ekraani sisenditele tuleb arvestada klaviatuuril tehtavate vajutustega. Tegemist on ajutise lahendusega, et õpilasel oleks võimalik enda tehtud mängu jooksutada ka siis, kui tal puudub puutetundliku ekraaniga seade.

Kolmanda peatüki eesmärk on siduda need kontrollid karakteriga. Selleks antakse alustuseks ülevaade eesmärgist, mida tahetakse saavutada. Seejärel lisatakse karakterile skript, mis võimaldab tal reageerida kasutaja käskudele. Esialgu luuakse üles ja alla liikumine. Peale seda lisatakse see skripti osa, mis tegeleb kuuli tulistamise loogikaga. Alustuseks kirjutatakse skripti see osa, mis loob uue kuuli objekti. Seejärel lisatakse kuulile trajektoor ning pannakse ta liikuma mööda trajektoori. Trajektoori implementeerimisel kirjeldatakse õpilasele algoritmi, mille põhjal see tehakse.

Kolmas peatükk näitab kasutajale, kuidas luua ja kasutada skripte. Need teadmised antakse edasi läbi praktilise näite, milleks on mängus oleva karakteri reageerimine sisendandmetele. Sisendandmeteks võivad olla kas klahvivajutused klaviatuuril või puudutused puutetundlikul ekraanil.

3.1.5 Mängu loogika ja füüsikasimulatsiooni tegemine

Neljanda teema alguseks on valmis loodud maailm ja sinna pandud juhitud karakter. Selleks, et karakteri poolt tulistatud kuulid purustaksid kokkupõrkel kastid, lisatakse mängu kokkupõrke kontrollija. Kontrollijaks on tööriist nimega Havok Physics.

Õppijale näidatakse, kuidas teha Havok Rigid Body objekti. Tegemist on mängu objektiga, millele saab külge panna füüsikalisi suuruseid nagu mass, tihedus ning mõõdud. Mõõtudeks on pikkus, laius ja kõrgus. Nende mõõtude ja objekti asukoha põhjal on võimalik selgeks teha, kas üks objekt on kokkupuutes mõne teise objektiga. Seda nimetatakse kokkupõrke kontrolliks.

Neljandas teemaplokis näidatakse õpilasele, kuidas käsitleda kokkupõrke kontrolle. Loodavas mängus on tarvis kokkupuuteid kontrollida vastavalt tabelis 3.1 olevale maatriksile. Kasutajale näidatakse, kuidas skriptides hallata kokkupõrgete kontrolli. Näiteks kuuli kokkupõrkamisel kastiga, on vaja mõlemad objektid mängumaailmast eemaldada. Lisaks võib näidata kasti purunemise animatsiooni.

Järgmise osana mängu loogika loengus luuakse punktide ja elude süsteem. Siin luuakse skript, mille ülesandeks on suurendada punkte, kui mängija tulistab objekti "boonus". Samas skriptis vähendatakse elupunkte, kui karakter on puutunud kokku mõne kastiga. Viimaseks osaks on mängu taaskäivitamine, kui järelejäänud elupunkte on null.

Neljanda loengu lõpuks on kasutajal valmis platvormmäng. Selle ehitamise käigus on kaetud põhilised aspektid mängude loomiseks Project Anarchy vahenditega. Kasutajale on näidatud objektide importimine, reageerimine sisenditele, kokkupõrgete haldamine ja skriptimine Lua keeles.

Kasutajal on võimalik peale loengut sooritada iseseisev teadmiste kontroll. Lisaks leidub loengute kõrval materjale, mis aitavad õpilasel leida rohkem infot käsitletud teemade kohta. Kursuse viimase teemana õpetatakse projekti eksportimist Androidi ja iOS operatsioonisüsteemidele.

3.1.6 Mängu testimine

Käesoleva kursuse eelnevate teemade ülesannetena on loodud mäng, mida tahetakse käivitada mobiilsetel platvormidel. Kursuse viies ehk viimane teema keskendubki projekti eksportimisele mängu loomise keskkonnast iOS ja Android süsteemile.

Kasutajale näidatakse vajalikke samme, et projekti saaks eksportida iOS ja Android platvormile. Seejärel käivitatakse eksporditud projekt seadmel ning testitakse. Testimisega on selle kursuse teemade ring ammendatud.

Õpilasele tehakse kokkuvõtte kõikidest loengutest. Sellega võetakse kogu kursus kokku. Lisaks jagatakse näpunäiteid, kust saada lisainformatsiooni Project Anarchy kohta. Lisainformatsiooni abil on õppijal võimalik oma mänguloomise oskusi edasi arendada väljaspool antud kursust.

Kursuse läbimise järel on õppija omandanud teadmised mängu loomise etappidest Project Anarchy vahenditega. Osaleja teab, kuidas alustada uue projektiga, importida sinna graafikaobjekte ning luua skriptidega mängu loogikat. Lisaks on kursuse materjalide seas viited teistele materjalidele, et oleks võimalik teemade kohta rohkem infot saada.

Käesolevas peatükis anti detailne ülevaade kursuse ehitusest. Kirjeldati iga teemat ning sellega seonduvaid ülesandeid. Materjalidele on võimalik ligi pääseda Udemy keskkonnas (vaata Lisa 1).

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli luua õppematerjalid rakendustarkvarale Project Anarchy. Õppematerjalid on koondatud kokku üheks grupiks, mille käigus valmib arvutimäng. Õppija omandab baasteadmised mängu loomisest kasutades Project Anarchy vahendeid ning oskab neid rakendada enda mängude tegemisel.

Töö koosneb kolmest peatükist, millest esimeses selgitati modernse mängumootori ülesehitust ja kirjeldati kahte mängumootorit. Üheks neist oli Unity ja teiseks Project Anarchy. Teises peatükis analüüsiti e-kursust ja selle ülesehitust. Määrati eeldused õppijale ning tutvustati Udemy veebiõppe keskkonda, kuhu loodavad materjalid üles laetakse. Kolmandas peatükis anti ülevaade loodava kursuse struktuurist. Kursuse teemasid selgitati detailselt, et lugejal oleks võimalik teada, mida oodata valmivast kursusest.

Bakalaureusetöö kirjutamise käigus sai autor laiendada oma silmaringi mängumootorite ehituse vallas ning rakendustarkvara Project Anarchy kasutamises. Lisaks omandas autor kursuse loomise kogemuse Udemy veebiõppekeskkonnas.

Viited

- [1] Fez (Video Game). [Online]. [http://en.wikipedia.org/wiki/Fez_\(video_game\)](http://en.wikipedia.org/wiki/Fez_(video_game)) - viimati vaadatud 13.05.2014
- [2] Game engine. [Online]. http://en.wikipedia.org/wiki/Game_engine - viimati vaadatud 13.05.2014
- [3] Unraveling the unending oink. [Online]. <http://unity3d.com/showcase/profiles/rovio-badpiggies> - viimati vaadatud 13.05.2014
- [4] Infinity Blade III. [Online]. http://en.wikipedia.org/wiki/Infinity_Blade_III - viimati vaadatud 13.05.2014
- [5] About Project Anarchy. [Online]. <https://www.projectanarchy.com/about> - viimati vaadatud 13.05.2014
- [6] Unity (game engine). [Online]. [http://en.wikipedia.org/wiki/Unity_\(game_engine\)#Versions](http://en.wikipedia.org/wiki/Unity_(game_engine)#Versions) - viimati vaadatud 13.05.2014
- [7] Mobile Hardware Stats. [Online]. <http://stats.unity3d.com/mobile/> - viimati vaadatud 13.05.2014
- [8] Assassin's Creed IV. [Online]. <http://assassinscreed.ubi.com/en-GB/home/index.aspx> - viimati vaadatud 13.05.2014
- [9] NFS Games. [Online]. <http://www.ea.com/nfs> - viimati vaadatud 13.05.2014
- [10] Nintendo's official home of Mario. [Online]. <http://mario.nintendo.com/> - viimati vaadatud 13.05.2014
- [11] Battlefield 4 (BF4). [Online]. <http://www.battlefield.com/battlefield-4> - viimati vaadatud 13.05.2014
- [12] Mobile operating system. [Online]. http://en.wikipedia.org/wiki/Mobile_operating_system - viimati vaadatud 13.05.2014
- [13] Cocos2D. [Online]. <http://www.cocos2d-iphone.org/> - viimati vaadatud 13.05.2014
- [14] Unity store. [Online]. <https://store.unity3d.com/> - viimati vaadatud 13.05.2014
- [15] Tutorials. [Online]. <http://unity3d.com/learn/tutorials/modules> - viimati vaadatud 13.05.2014
- [16] Project Anarchy. [Online]. <http://projectanarchy.com/> - viimati vaadatud 13.05.2014
- [17] Publication: Games King of the Mobile Eco-system
http://www.distimo.com/blog/2013_10_publication-games-king-of-the-mobile-eco-system/ - viimati vaadatud 13.05.2014
- [18] Bad Piggies ekraanitõmmis. [Online]. http://17f0418678386b4e6860-e4f9fcd924b589d19bf6ccc2802ea9aa.r66.cf1.rackcdn.com/7dc436e36c6b33d830409105da2628a71785a053.jpg_0x529_q85_upscale.jpg - viimati vaadatud 13.05.2014
- [19] Infinity Blade ekraanitõmmis. [Online]. <http://i.imgur.com/xKEz1t2.jpg> - viimati vaadatud 13.05.2014
- [20] Jason Gregory, Jeff Lander, Matt Whiting, "Game Engine Architecture," vol. 1, pp 28-57, 2009.

- [21] Features in iOS. [Online]. <https://developer.apple.com/technologies/ios/features.html> - viimati vaadatud 13.05.2014
- [22] iOS App Programming Guide. [Online]. https://developer.apple.com/library/ios/documentation/iphone/conceptual/iphoneosprogrammingguide/ManagingYourApplicationsFlow/ManagingYourApplicationsFlow.html#//apple_ref/doc/uid/TP40007072-CH4-SW3 - viimati vaadatud 13.05.2014
- [23] Vallaste e-teatmik- utiliit. [Online]. <http://www.vallaste.ee/index.htm?Type=UserId&otsing=%20619> - viimati vaadatud 13.05.2014
- [24] Havok Releases Project Anarchy Completely Free. [Online]. <http://www.projectanarchy.com/news/havok-releases-project-anarchy-completely-free> - viimati vaadatud 13.05.2014
- [25] Licensing. [Online]. <https://projectanarchy.com/licensing> - viimati vaadatud 13.05.2014
- [26] Tizen. [Online]. <https://www.tizen.org/> - viimati vaadatud 13.05.2014
- [27] Scripting 1. [Online]. <http://www.projectanarchy.com/scripting-1> - viimati vaadatud 13.05.2014
- [28] Anarchy RPG Appstore. [Online]. <https://itunes.apple.com/us/app/anarchy-rpg/id660485296?mt=8> - viimati vaadatud 13.05.2014
- [29] Anarchy RPG Google Play. [Online]. [\https://play.google.com/store/apps/details?id=com.havok.Vision.AnarchyRPG - viimati vaadatud 13.05.2014
- [30] Unity - Multiplatform. [Online]. <https://unity3d.com/unity/multiplatform> - viimati vaadatud 13.05.2014
- [31] Unity - System Requirements. [Online]. <http://unity3d.com/unity/system-requirements> - viimati vaadatud 13.05.2014
- [32] Unity - Animation. [Online]. <https://unity3d.com/unity/animation> - viimati vaadatud 13.05.2014
- [33] Unity - Store. [Online]. <https://store.unity3d.com/> - viimati vaadatud 13.05.2014
- [34] Unity - Scripting. [Online]. <https://unity3d.com/learn/tutorials/modules/beginner/scripting> - viimati vaadatud 13.05.2014
- [35] Kerbal Space Program. [Online]. <https://kerbalspaceprogram.com/> - viimati vaadatud 13.05.2014
- [36] Blackguards. [Online]. <http://www.blackguards-game.com/de/> - viimati vaadatud 13.05.2014
- [37] About Udemy. [Online]. <https://www.udemy.com/about/> - viimati vaadatud 13.05.2014
- [38] Lua (programming language). [Online]. [http://en.wikipedia.org/wiki/Lua_\(programming_language\)](http://en.wikipedia.org/wiki/Lua_(programming_language)) - viimati vaadatud 13.05.2014
- [39] Infinity Blade III. [Online]. <http://www.metacritic.com/game/ios/infinity-blade-iii> - viimati vaadatud 13.05.2014
- [40] Juhend kvaliteetse e-kursuse loomiseks. Versioon 2.0. [Online]. http://www.e-ope.ee/_download/repository/Kasiraamat2.0.pdf - viimati vaadatud 13.05.2014
- [41] Udemy Support. [Online].

<http://support.udemy.com/knowledgebase/articles/159344-when-will-udemy-promote-my-course>
- viimati vaadatud 13.05.2014

Lisad

Lisa 1. Kursuse materjalid

Kursuse raames loodud õppematerjalide ligipääsuks on loodud Udemy keskkonda privaatne kursus. Sellele ligipääsemiseks peaks kontakteeruma töö autoriga, kes tagab ligipääsu kursusele.

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina **Timo Kallaste** (sünnikuupäev: 22.03.1989)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose **Õppematerjalide loomine rakendustarkvarale Project Anarchy**, mille juhendajad on Anne Villems ja Taavi Duvin,

- 1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
- 1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **13.05.2014**